Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/005136

International filing date: 22 March 2005 (22.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-107778
Filing date: 31 March 2004 (31.03.2004)

Date of receipt at the International Bureau: 28 April 2005 (28.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 Date of Application:	２００４年　３月３１日
出 願 番 号 Application Number:	特願２００４－１０７７７８
パリ条約による外国への出願に用いる優先権の主張の基礎となる出願の国コードと出願番号 The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is	ＪＰ２００４－１０７７７８
出 願 人 Applicant(s):	松下電器産業株式会社

２００５年　４月１３日

特許庁長官
Commissioner,
Japan Patent Office

小 川

【書類名】　特許願
【整理番号】　2048160031
【提出日】　平成16年 3月31日
【あて先】　特許庁長官　殿
【国際特許分類】　G09C　5/00
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　布田　裕一
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　山道　将人
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　大森　基司
【発明者】
　【住所又は居所】　宮城県仙台市青葉区上杉５－８－７－６０６
　【氏名】　静谷　啓樹
【発明者】
　【住所又は居所】　宮城県仙台市青葉区星陵町３－３６－２０３
　【氏名】　満保　雅浩
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100090446
　【弁理士】
　【氏名又は名称】　中島　司朗
【手数料の表示】
　【予納台帳番号】　014823
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9003742

【書類名】特許請求の範囲
【請求項１】
群Ｇ用いて、２個以上の整数の加算を行う加算プログラムであって、
前記群Ｇは、真に含む部分群Ｓをもち、
前記群Ｇ上の冪演算を行うことにより、前記整数を前記Ｇに属する元に変換する変換モジュールと、
前記群Ｇ上の基本演算を行う主要演算モジュールと、
前記群Ｇまたは、前記群Ｇの部分群Ｓにおける前記変換モジュールで行う変換の逆変換を行う逆変換モジュールと、を備えることを特徴とする、加算プログラム。
【請求項２】
前記群Ｇは、剰余整数環の乗法群であることを特徴とする、請求項１記載の加算プログラム。
【請求項３】
前記群Ｇは、複数個の相異なる素数ｐ１，ｐ２，…，ｐｋ（ｋ＞１）の積ｎ＝ｐ１×ｐ２×…×ｐｋに対し、Ｚ／ｎＺの乗法群であることを特徴とする、請求項２記載の加算プログラム（ただし、×は乗算である。また、Ｚは整数環であり、Ｚ／ｎＺは整数をｍｏｄｎした値で構成される剰余整数環である）。
【請求項４】
前記逆変換モジュールは、前記素数ｐ１，ｐ２，…，ｐｋを用いたＺ／ｐ１　Ｚ，Ｚ／ｐ２　Ｚ，…，Ｚ／ｐｋ　Ｚの乗法群における離散対数問題を解くことを特徴とする、請求項３記載の加算プログラム。
【請求項５】
前記逆変換モジュールは、前記素数ｐ１，ｐ２，…，ｐｋを用いたＺ／ｐ１　Ｚ，Ｚ／ｐ２　Ｚ，…，Ｚ／ｐｋ　Ｚの乗法群における離散対数問題の解に対し、中国人の剰余定理を用いることを特徴とする、請求項４記載の加算プログラム。
【請求項６】
前記群Ｇは、２つの素数ｐ，ｑと正整数ｍを用いて表されるｎ＝ｐ＾ｍ×ｑに対し、Ｚ／ｎＺの乗法群であることを特徴とする、請求項１記載の加算プログラム（ただし、ｘ＾ｙはｘのｙ乗を示す）。
【請求項７】
前記部分群Ｓは、Ｚ／ｐ＾ｍ　Ｚの乗法群であることを特徴とする、請求項６記載の加算プログラム。
【請求項８】
前記正整数ｍは２であることを特徴とする、請求項６記載の加算プログラム。
【請求項９】
前記部分群Ｓは、アノマラス楕円曲線の群であることを特徴とする、請求項１記載の加算プログラム。
【請求項１０】
前記群Ｇは、二つのアノマラス楕円曲線の群の直積であることを特徴とする、請求項１記載の加算プログラム。
【請求項１１】
さらに、前記逆変換モジュールは、予め一種類以上の数を冪乗もしくは冪倍して冪演算した結果を格納した格納手段を備えることを特徴とする、請求項１から請求項１０のいずれか１項に記載の加算プログラム。
【請求項１２】
さらに、前記逆変換モジュールは、前記群Ｇに属する元を前記部分群Ｓに属する元に還元する還元手段を備えることを特徴とする、請求項１から請求項１０のいずれか１項に記載の加算プログラム。
【請求項１３】
平文と鍵から暗号文を計算する暗号化プログラムであって、

整数同士の加算を行う一以上の加算モジュールを備え、
前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする、暗号化プログラム。
【請求項１４】
前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項１３記載の暗号化プログラム。
【請求項１５】
前記暗号文は共通鍵暗号を用いて計算することを特徴とする、請求項１３または請求項１４記載の暗号化プログラム。
【請求項１６】
暗号文と鍵から平文を計算する復号化プログラムであって、
整数同士の加算を行う一以上の加算モジュールを備え、
前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする、復号化プログラム。
【請求項１７】
前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項１６記載の復号化プログラム。
【請求項１８】
前記平文は共通鍵暗号を用いて計算することを特徴とする、請求項１６または請求項１７記載の復号化プログラム。
【請求項１９】
データに対し、デジタル署名を生成する署名生成プログラムであって、
整数同士の加算を行う一以上の加算モジュールを備え、
前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする、署名生成プログラム。
【請求項２０】
前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項１９記載の署名生成プログラム。
【請求項２１】
平文と鍵から暗号文を計算する暗号化装置であって、
整数同士の加算を行う一以上の加算演算部を備え、
前記加算演算部の一部または全部を請求項１から請求項１２記載の加算プログラムを処理して実行することを特徴とする、暗号化装置。
【請求項２２】
前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項２１記載の暗号化装置。
【請求項２３】
暗号文と鍵から平文を計算する復号化装置であって、
整数同士の加算を行う一以上の加算演算部を備え、
前記加算演算部の一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする、復号化装置。
【請求項２４】
前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする、請求項２３記載の復号化装置。
【請求項２５】
群Ｇ用いて、２個以上の整数の加算を行う加算方法であって、
前記群Ｇは、真に含む部分群Ｓをもち、
前記群Ｇ上の冪演算を行うことにより、前記整数を前記Ｇに属する元に変換する変換ステップと、
前記群Ｇ上の基本演算を行う主要演算ステップと、

前記群Ｇまたは、前記群Ｇの部分群Ｓにおける前記変換ステップで行う変換の逆変換を行う逆変換ステップと、を含むことを特徴とする、加算方法。

【請求項２６】

請求項１から請求項２０のいずれか１項に記載のプログラムを記録した記録媒体。

【請求項２７】

請求項１から請求項２０のいずれか１項に記載のプログラムを実行するＩＣカード。

【書類名】明細書
【発明の名称】加算プログラム
【技術分野】
【０００１】
本発明は、情報セキュリティ技術としてのソフトウェアの難読化方法に関する。
【背景技術】
【０００２】
暗号ソフトを実装する場合、鍵や暗号アルゴリズムをそのまま実装すると、ソフトを解析された場合に、簡単に不正使用できる。そのため、ソフト解析を困難にする耐タンパーソフト技術が要望されている。耐タンパーソフト技術として、特許文献１において、演算及び演算領域を変換して、変換前の演算領域を推測困難にすることにより、ソフト解析困難とする方式が記載されている。
【０００３】
その方式は、一次変換などにより変換を行う。例えば、鍵とデータの加算を変換する場合、鍵とデータを変換し、変換後のデータを変換後の領域における加算を行い、その結果に逆変換を実行することにより鍵とデータの加算結果を得る。このような難読化された加算方法を、共通鍵暗号を用いた暗号化プログラムや復号化プログラムに適用することにより、プログラムを解析して鍵を得る攻撃に対する安全性を向上させることが可能になる。
【０００４】
〔従来例〕
図６は、加算プログラム５００の構成を示す図である。本プログラムは変換モジュール５１０、主要演算モジュール５２０、逆変換モジュール５３０、出力モジュール５４０から構成される。本プログラムは、入力ａ，ｂに対し、ａ＋ｂを出力するものである。
変換モジュール５１０は、整数ｋ１，ｋ２を保持し、これらを用いて、入力ａ，ｂをそれぞれｔａ＝ｋ１×ａ＋ｋ２，ｔｂ＝ｋ１×ｂ＋ｋ２に変換する。ただし、×は乗算を示す。
次に主要演算モジュール５２０は、ｔａ，ｔｂに対して、ｔａｂ＝ｔａ＋ｔｂを計算する。
逆変換モジュール５３０は、ｔａｂに対して、ｃ＝（ｔａｂ－２ｋ２）／ｋ１を計算する。
出力モジュール５４０は、ｃを出力する。
【０００５】
上記のように処理すると、ｔａｂ＝ｔａ＋ｔｂ＝ｋ１×ａ＋ｋ２＋ｋ１×ｂ＋ｋ２＝ｋ１×（ａ＋ｂ）＋２ｋ２より、（ｔａｂ－２ｋ２）／ｋ１＝ａ＋ｂが成り立つ。したがって、ｃ＝ａ＋ｂとなり、加算プログラム５００によりａとｂの加算結果が得られる。
ここで、変換モジュール５１０及び逆変換モジュール５３０を解析困難なように実現した場合、解析者が解析可能なものは、ｔａ，ｔｂ，ｔａｂのみであり、これらの値からａ，ｂを推測するのが難しいため、ａ，ｂを隠蔽することが可能になる。
【特許文献１】米国特許第６５９４７６１号明細書
【特許文献２】特許第３４０２４４１号明細書
【特許文献３】特許第２７６０７９９号明細書
【非特許文献１】岡本龍明、山本博資、「現代暗号」、産業図書（１９９７年）
【非特許文献２】Ｈｅｎｒｉ Ｃｏｈｅｎ，“Ａ Ｃｏｕｒｓｅ ｉｎ Ｃｏｍｐｕｔａｔｉｏｎａｌ Ａｌｇｅｂｒａｉｃ Ｎｕｍｂｅｒ Ｔｈｅｏｒｙ”，ＧＴＭ １３８，Ｓｐｒｉｎｇｅｒ－Ｖｅｒｌａｇ，１９９３，ｐｐ．１９－２０
【非特許文献３】Ｉ．Ｂｌａｋｅ，Ｇ．Ｓｅｒｏｕｓｓｉ ａｎｄ Ｎ．Ｓｍａｒｔ，“Ｅｌｌｉｐｔｉｃ Ｃｕｒｖｅｓ ｉｎ Ｃｒｙｐｔｏｇｒａｐｈｙ”，ＣＡＭＢＲＩＤＧＥ ＵＮＩＶＥＲＳＩＴＹ ＰＲＥＳＳ，１９９９
【非特許文献４】Ｎ．Ｋｕｎｉｈｉｒｏ ａｎｄ Ｋ．Ｋｏｙａｍａ，“Ｔｗｏ Ｄｉｓｃｒｅｔｅ Ｌｏｇ Ａｌｇｏｒｉｔｈｍｓ ｆｏｒ Ｓｕｐｅｒ－Ａｎ

omalous Elliptic Curves”, SCIS’99, 1999, pp. 869－874

【発明の開示】

【発明が解決しようとする課題】

【０００６】

上記従来例の方式は、変換後の領域においても通常の加算と同様の演算を行うため、変換後の演算を解析することにより、変換前の演算が加算であることが判明しやすいという課題がある。プログラムの基となるアルゴリズムが解析者に既知の場合、解析者が加算をしている場所を分かると、その部分に集中して解析することにより、どのような変換を行っているか知られてしまう恐れがある。ゆえに、可能な限り、変換前の演算を知られないようにした方がよい。

【０００７】

本発明は、変換前の演算を他の演算に変えるような変換を行うことにより、変換前の演算を解析しにくくするソースコード難読化方法を提供することを目的とする。

【課題を解決するための手段】

【０００８】

上記目的を達成するために、請求項１における発明は、群Ｇ用いて、２個以上の整数の加算を行う加算プログラムであって、前記群Ｇは、真に含む部分群Ｓをもち、前記群Ｇ上の冪演算を行うことにより、前記整数を前記Ｇに属する元に変換する変換モジュールと、前記群Ｇ上の基本演算を行う主要演算モジュールと、前記群Ｇまたは、前記群Ｇの部分群Ｓにおける前記変換モジュールで行う変換の逆変換を行う逆変換モジュールと、を備えることを特徴とする。

【０００９】

請求項２における発明は、前記群Ｇは、剰余整数環の乗法群であることを特徴とする。

請求項３における発明は、前記群Ｇは、複数個の相異なる素数ｐ１，ｐ２，…，ｐｋ（ｋ＞１）の積ｎ＝ｐ１×ｐ２×…×ｐｋに対し、Ｚ／ｎＺの乗法群であることを特徴とする（ただし、×は乗算である。また、Ｚは整数環であり、Ｚ／ｎＺは整数をｍｏｄ　ｎした値で構成される剰余整数環である）。

【００１０】

請求項４における発明は、前記逆変換モジュールは、前記素数ｐ１，ｐ２，…，ｐｋを用いたＺ／ｐ１　Ｚ，Ｚ／ｐ２　Ｚ，…，Ｚ／ｐｋ　Ｚの乗法群における離散対数問題を解くことを特徴とする。

請求項５における発明は、前記逆変換モジュールは、前記素数ｐ１，ｐ２，…，ｐｋを用いたＺ／ｐ１　Ｚ，Ｚ／ｐ２　Ｚ，…，Ｚ／ｐｋ　Ｚの乗法群における離散対数問題の解に対し、中国人の剰余定理を用いることを特徴とする。

【００１１】

請求項６における発明は、前記群Ｇは、２つの素数ｐ，ｑと正整数ｍを用いて表されるｎ＝ｐ＾ｍ×ｑに対し、Ｚ／ｎＺの乗法群であることを特徴とする。（ただし、ｘ＾ｙはｘのｙ乗を示す。）

請求項７における発明は、前記部分群Ｓは、Ｚ／ｐ＾ｍ　Ｚの乗法群であることを特徴とする。

【００１２】

請求項８における発明は、前記正整数ｍは２であることを特徴とする。

請求項９における発明は、前記部分群Ｓは、アノマラス楕円曲線の群であることを特徴とする。

請求項１０における発明は、前記群Ｇは、二つのアノマラス楕円曲線の群の直積であることを特徴とする。

【００１３】

請求項１１における発明は、さらに、前記逆変換モジュールは、予め一種類以上の数を冪乗もしくは冪倍して冪演算した結果を格納した格納手段を備えることを特徴とする。

請求項１２における発明は、さらに、前記逆変換モジュールは、前記群Ｇに属する元を前記部分群Ｓに属する元に還元する還元手段を備えることを特徴とする。
請求項１３における発明は、平文と鍵から暗号文を計算する暗号化プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする。
【００１４】
請求項１４における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。
請求項１５における発明は、前記暗号文は共通鍵暗号を用いて計算することを特徴とする。
請求項１６における発明は、暗号文と鍵から平文を計算する復号化プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする。
【００１５】
請求項１７における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。
請求項１８における発明は、前記平文は共通鍵暗号を用いて計算することを特徴とする。
請求項１９における発明は、データに対し、デジタル署名を生成する署名生成プログラムであって、整数同士の加算を行う一以上の加算モジュールを備え、前記加算モジュールの一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする。
【００１６】
請求項２０における発明は、前記加算モジュールは、鍵の一部または全部と整数との加算を行うことを特徴とする。
請求項２１における発明は、平文と鍵から暗号文を計算する暗号化装置であって、整数同士の加算を行う一以上の加算演算部を備え、前記加算演算部の一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする。
【００１７】
請求項２２における発明は、前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする。
請求項２３における発明は、暗号文と鍵から平文を計算する復号化装置であって、整数同士の加算を行う一以上の加算演算部を備え、前記加算演算部の一部または全部を請求項１から請求項１２のいずれか１項に記載の加算プログラムを処理して実行することを特徴とする。
【００１８】
請求項２４における発明は、前記加算演算部は、鍵の一部または全部と整数との加算を行うことを特徴とする。
請求項２５における発明は、群Ｇ用いて、２個以上の整数の加算を行う加算方法であって、前記群Ｇは、真に含む部分群Ｓをもち、前記群Ｇ上の冪演算を行うことにより、前記整数を前記Ｇに属する元に変換する変換ステップと、前記群Ｇ上の基本演算を行う主要演算ステップと、前記群Ｇまたは、前記群Ｇの部分群Ｓにおける前記変換ステップで行う変換の逆変換を行う逆変換ステップと、を含むことを特徴とする。
【００１９】
請求項２６における発明は、請求項１から請求項２０のいずれか１項に記載のプログラムを記録した記録媒体である。
請求項２７における発明は、請求項１から請求項２０のいずれか１項に記載のプログラ

ムを実行するＩＣカードである。
【発明の効果】
【００２０】
これらの構成によると、演算に使用する値の隠蔽だけでなく、演算そのものを隠蔽することができ、その価値は大きい。
【発明を実施するための最良の形態】
【００２１】
〔実施の形態１〕
本発明にかかる実施の形態１としての加算プログラム１００について説明する。
図１は、実施の形態１における加算プログラム１００の構成を示す図である。本プログラムは、変換モジュール１１０、主要演算モジュール１２０、逆変換モジュール１３０、出力モジュール１４０から構成される。本プログラムは、入力ａ，ｂに対し、ａ＋ｂを出力するものである。
【００２２】
＜各種パラメータ・記号の定義・入力の条件＞
ここで、実施の形態１としての加算プログラム１００で使用する各種パラメータ・記号の定義を示す。
ｐｉ（ｉ＝１，２，…，ｋ）を素数とし、それらの積ｐ１×ｐ２×…×ｐｋをｎとする。ここで、×は乗算を示す。ｐｉ（ｉ＝１，２，…，ｋ）は逆変換モジュール１３０が保持し、ｎは変換モジュール１１０、主要演算モジュール１２０がそれぞれ保持する。
【００２３】
実施の形態１では、ｍｏｄ　ｎの整数から構成される剰余整数環Ｚ／ｎＺの乗法群の演算を用いる。ｇは、その乗法群に属する予め与えられた数であり、ｐｉ（ｉ＝１，２，…，ｋ）に対して原始元とする。原始元とは、ｍを１，　２，　…と動かしたとき、初めてｇ＾ｍ＝１　ｍｏｄ　ｐｉとなるｍの値がｐｉ－１であるようなｇである。
Ｌ＝ｌｃｍ（ｐ１－１，ｐ２－１，…，ｐｋ－１）とする。ここで、ｌｃｍ（ｐ１－１，ｐ２－１，…，ｐｋ－１）はｐ１－１，ｐ２－１，…，ｐｋ－１の最小公倍数を示す。
【００２４】
入力ａ，ｂは、Ｌ／２より小さい非負整数とする。
＜変換モジュール１１０の構成・処理＞
変換モジュール１１０は、ｎとｇを格納する第１のパラメータ格納手段と、入力ａ，ｂに対し、ｇａ＝ｇ＾ａ　ｍｏｄ　ｎ，ｇｂ＝ｇ＾ｂ　ｍｏｄ　ｎを計算する冪乗計算手段を備える。ここで、ｇ＾ａはｇのａ乗を示す。
【００２５】
変換モジュール１１０は、入力ａ，ｂに対し、ｇａ＝ｇ＾ａ　ｍｏｄ　ｎ，ｇｂ＝ｇ＾ｂ　ｍｏｄ　ｎを計算し、それらの計算結果ｇａ，ｇｂを主要演算モジュール１２０へ入力する。
＜主要演算モジュール１２０の構成・処理＞
主要演算モジュール１２０は、ｎを格納する第２のパラメータ格納手段と、入力ｇａ，ｇｂに対し、ｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算する乗算手段を備える。
【００２６】
主要演算モジュール１２０は、入力ｇａ，ｇｂに対し、第２のパラメータ格納手段に格納されているｎを用いて、ｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算し、その計算結果ｇａｂを逆変換モジュール１３０へ入力する。
＜逆変換モジュール１３０の構成・処理＞
逆変換モジュール１３０は、ｐｉ（ｉ＝１，２，…，ｋ）を格納する第３のパラメータ格納手段と、入力ｇａｂに対し、ｇａｂ　ｍｏｄ　ｐｉ（ｉ＝１，２，…，ｋ）のｇ　ｍｏｄ　ｐｉに対する離散対数ｃｉ　ｍｏｄ　ｐｉ－１を計算する離散対数計算手段と、離散対数計算手段で計算したｉ＝１，２，…，ｋに対する各ｃｉから、ｇａｂ　ｍｏｄ　ｎのｇ　ｍｏｄ　ｎに対する離散対数ｃ　ｍｏｄ　Ｌを求める中国人剰余定理利用手段を備

える。
【００２７】
逆変換モジュール１３０は、まず、入力ｇａｂに対し、ｉ＝１，２，…，ｋに対して、ｇａｂ　ｍｏｄ　ｐｉのｇ　ｍｏｄ　ｐｉに対する離散対数ｃｉ　ｍｏｄ　ｐｉ－１を計算する。次に、ｉ＝１，２，…，ｋに対する各ｃｉを用いて、中国人の剰余定理（Ｃｈｉｎｅｓｅ　Ｒｅｍａｉｎｄｅｒ　Ｔｈｅｏｒｅｍ）を用いて、ｃ　ｍｏｄ　Ｌを計算し、ｃを出力モジュール１４０へ入力する。
【００２８】
離散対数計算手段におけるｃｉ　ｍｏｄ　ｐｉ－１の計算方法については、様々なものがあるが、以下でその一例を示す。
ｗを１，２，３，…と順番に動かして、ｇ＾ｗ＝ｇａｂ　ｍｏｄ　ｐｉとなるｗを求める。そのｗをｃｉとする。なお、各ｐｉに関し、ｇ＾１，ｇ＾２，…，ｇ＾（ｐｉ－２）ｍｏｄ　ｐｉを計算した結果をテーブルとして保持し、そのテーブルの値から、ｇａｂ　ｍｏｄ　ｐｉと一致するｇ＾ｗを探す、としてもよい。
【００２９】
中国人剰余定理利用手段におけるｃｉ（ｉ＝１，２，．．．，ｋ）から，ｃ　ｍｏｄ　ｐｉ－１＝ｃｉを満たすｃ　ｍｏｄ　Ｌを計算する方法については、非特許文献２が詳しい。
＜出力モジュール１４０の構成・処理＞
出力モジュール１４０は、プログラムの外部へ数値を出力する出力手段を備える。
【００３０】
出力モジュール１４０は、入力ｃをプログラムの外部へ出力する。
＜実施の形態１の全体の動作＞
実施の形態１における加算プログラム１００の全体の動作を以下に示す。
加算プログラム１００の変換モジュール１１０は、入力ａ，ｂに対しｇａ，ｇｂを計算する。次に主要演算モジュール１２０は、ｇａ，ｇｂに対し、ｇａｂを計算する。さらに、逆変換モジュール１３０は、ｇａｂに対し、ｃを計算し、出力モジュール１４０は、ｃを出力する。
【００３１】
＜実施の形態１の動作検証＞
以下で、実施の形態１における加算プログラム１００が、入力ａ，ｂに対し、ａ＋ｂを出力していることを検証する。
変換モジュール１１０において、ａ，ｂに対し、ｇａ＝ｇ＾ａ　ｍｏｄ　ｎ，ｇｂ＝ｇ＾ｂ　ｍｏｄ　ｎを計算し、主要演算ステップモジュール１２０でｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算する。このとき、ｇａｂ＝ｇ＾（ａ＋ｂ）　ｍｏｄ　ｎを満たすことは明らかである。逆変換モジュール１３０では、ｇとｇａｂからｇａｂ＝ｇ＾ｃｉ　ｍｏｄ　ｐｉ（ｉ＝１，２，…，ｋ）を満たすｃｉを計算し、その結果を用いて、ｃ＝ｃｉ　ｍｏｄ　ｐｉ－１を満たすｃ　ｍｏｄ　Ｌを計算する。このとき、ｃはｇａｂ＝ｇ＾ｃ　ｍｏｄ　ｎを満たす。なぜなら、ａ＋ｂ＝ｃ　ｍｏｄ　Ｌより、ｇ＾（ａ＋ｂ－ｃ）＝１　ｍｏｄ　ｎとなるためである。したがって、ｇ＾（ａ＋ｂ）　ｍｏｄ　ｎ＝ｇ＾ｃ　ｍｏｄ　ｎを満たすため、ａ＋ｂ＝ｃ　ｍｏｄ　（（ｐ１－１）×（ｐ２－１）×…×（ｐｋ－１））を満たす。ａ＜Ｌ／２，ｂ＜Ｌ／２よりａ＋ｂ＜Ｌであるので、加算プログラム１００はａとｂの加算結果ａ＋ｂを出力していることになる。
【００３２】
＜実施の形態１の効果＞
実施の形態１における加算プログラム１００は、加算を行う値を変換しており、変換モジュール１１０及び逆変換モジュール１３０が解析困難な場合に、解析者はｇａ，ｇｂ，ｇａｂの値と、ｇａ，ｇｂからｇａｂを計算する処理を解析可能である。そのとき、変換後の値ｇａ，ｇｂから変換前の値ａ，ｂを推測することは困難である。さらに、加算プログラム１００は、主要演算モジュール１２０において、乗算を行っており、この乗算とい

う演算から加算プログラム１００が加算を実現していることを推測することは困難である。したがって、加算を行う入力の値の隠蔽だけではなく、加算という演算自体も隠蔽できることになり、本実施の形態１は有効である。

【００３３】

〔実施の形態２〕

本発明にかかる実施の形態２としての加算プログラム２００について説明する。

図２は、実施の形態２における加算プログラム２００の構成を示す図である。本プログラムは、実施の形態１と同様の出力モジュール１４０と、実施の形態１と異なる変換モジュール２１０、主要演算モジュール２２０、逆変換モジュール２３０から構成される。本プログラムは、入力ａ，ｂに対し、ａ＋ｂを出力するものである。

【００３４】

＜各種パラメータ・記号の定義・入力の条件＞

ここで、実施の形態２としての加算プログラム２００で使用する各種パラメータ・記号の定義を示す。

ｐ，ｑを素数とし、ｎ＝ｐ＾２×ｑとする。ｐ，ｑは逆変換モジュール２３０が保持し、ｎは変換モジュール２１０、主要演算モジュール２２０がそれぞれ保持する。

【００３５】

実施の形態２では、ｍｏｄ　ｎの整数から構成される剰余整数環Ｚ／ｎＺの乗法群の演算を用いる。ｇは、その乗法群に属する予め与えられた数でありｇ＾（ｐ－１）　ｍｏｄ　ｐ＾２の位数がｐである数とする。ｇｐ＝ｇ＾（ｐ－１）　ｍｏｄ　ｐ＾２と定義する。

入力ａ，ｂは、ｐ／２より小さい非負整数とする。

【００３６】

＜変換モジュール２１０の構成・処理＞

変換モジュール２１０は、ｎとｇを格納する第１のパラメータ格納手段と、ｎ以下の乱数Ｒ１，Ｒ２を発生する乱数発生手段と、入力ａ，　ｂに対し、乱数発生手段が発生した乱数Ｒ１，Ｒ２を用いて、ｇａ＝ｇ＾（ａ＋ｎ×Ｒ１）　ｍｏｄ　ｎ，ｇｂ＝ｇ＾（ｂ＋ｎ×Ｒ２）　ｍｏｄ　ｎを計算する冪乗計算手段を備える。

【００３７】

変換モジュール２１０は、乱数Ｒ１，Ｒ２を発生して、入力ａ，ｂに対し、ｇａ＝ｇ＾（ａ＋ｎ×Ｒ１）　ｍｏｄ　ｎ，　ｇｂ＝ｇ＾（ｂ＋ｎ×Ｒ２）　ｍｏｄ　ｎを計算し、それらの計算結果ｇａ，ｇｂを主要演算モジュール２２０へ入力する。

＜主要演算モジュール２２０の構成・処理＞

主要演算モジュール２２０は、ｎを格納する第２のパラメータ格納手段と、入力ｇａ，ｇｂに対し、ｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算する乗算手段を備える。

【００３８】

主要演算モジュール２２０は、入力ｇａ，ｇｂに対し、第２のパラメータ格納手段に格納されているｎを用いて、ｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算し、その計算結果ｇａｂを逆変換モジュール２３０へ入力する。

＜逆変換モジュール２３０の構成・処理＞

逆変換モジュール２３０は、ｐを格納する第３のパラメータ格納手段と、入力ｇａｂに対し、第３のパラメータ格納手段に格納されているｐを用いて、ｃｐ＝ｇａｂ＾（ｐ－１）　ｍｏｄ　ｐ＾２を計算する還元手段と、ｍｏｄ　ｐ＾２におけるｇｐに対するｃｐの離散対数を計算する離散対数計算手段を備える。

【００３９】

逆変換モジュール２３０は、まず、入力ｇａｂに対し、ｃｐ＝ｇａｂ＾（ｐ－１）　ｍｏｄ　ｐ＾２を計算する。次に、ｍｏｄ　ｐ＾２におけるｇｐに対するｃｐの離散対数ｃを計算し、ｃを出力モジュール２４０へ入力する。

離散対数計算手段におけるｃの計算方法は、特許文献２が詳しい。

具体的には、以下のように行う。

【００４０】
ｃｐに対し、ｃ＝（ｃｐ－１）／（ｇ－１）　ｍｏｄ　ｐとしてｃを求める。
＜実施の形態２の全体の動作＞
実施の形態２における加算プログラム２００の全体の動作を以下に示す。
加算プログラム２００の変換モジュール２１０は、入力ａ，ｂに対しｇａ，ｇｂを計算する。次に主要演算モジュール２２０は、ｇａ，ｇｂに対し、ｇａｂを計算する。さらに、逆変換モジュール２３０は、ｇａｂに対し、ｃを計算し、出力モジュール１４０は、ｃを出力する。
【００４１】
＜実施の形態２の動作検証＞
以下で、実施の形態２における加算プログラム２００が、入力ａ，ｂに対し、ａ＋ｂを出力していることを検証する。
変換モジュール２１０において、ａ，ｂに対し、ｇａ＝ｇ＾（ａ＋ｎ×Ｒ１）　ｍｏｄ　ｎ，ｇｂ＝ｇ＾（ｂ＋ｎ×Ｒ２）　ｍｏｄ　ｎを計算し、主要演算ステップモジュール２２０でｇａｂ＝ｇａ×ｇｂ　ｍｏｄ　ｎを計算する。このとき、ｇａｂ＝ｇ＾（ａ＋ｂ＋ｎ×（Ｒ１＋Ｒ２））　ｍｏｄ　ｎを満たすことは明らかである。逆変換モジュール２３０では、まず、ｃｐ＝ｇａｂ＾（ｐ－１）＝ｇｐ＾（ａ＋ｂ＋ｎ×（Ｒ１＋Ｒ２））　ｍｏｄ　ｐ＾２となり、ｇｐ＾ｐ＝１　ｍｏｄ　ｐ＾２より、ｇｐ＾ｎ＝１　ｍｏｄ　ｐ＾２であるため、ｃｐ＝ｇｐ＾（ａ＋ｂ）　ｍｏｄ　ｐ＾２となる。
【００４２】
逆変換モジュール２３０では、さらに、ｍｏｄ　ｐ＾２におけるｇｐに対するｃｐの離散対数ｃを求める。すなわち、ｃｐ＝ｇｐ＾ｃ　ｍｏｄ　ｐ＾２が成り立つ。したがって、ｃ＝ａ＋ｂ　ｍｏｄ　ｐであり、ａ＜ｐ／２，ｂ＜ｐ／２より、ａ＋ｂ＜ｐであるため、加算プログラム２００はａとｂの加算結果ａ＋ｂを出力していることになる。
＜実施の形態２の効果＞
実施の形態２における加算プログラム２００は、実施の形態１と同様に、加算を行う値を変換しており、変換モジュール２１０及び逆変換モジュール２３０が解析困難な場合に、変換後の値から変換前の値を推測することは困難である。さらに、加算プログラム２００は、主要演算モジュール２２０において、乗算を行っており、この乗算という演算から加算プログラム２００が加算を実現していることを推測することは困難である。したがって、加算を行う入力の値の隠蔽だけではなく、加算という演算自体も隠蔽できることになり、本実施の形態２は有効である。
【００４３】
実施の形態２における加算プログラム２００では、剰余整数環Ｚ／ｎＺの乗法群における冪乗演算を変換モジュールで行い、その乗法群の部分群である剰余整数環Ｚ／ｐ＾２Ｚの乗法群における離散対数問題を逆変換モジュールで解いている。ここで、もし、解析者がｐ，ｑは分からないが、変換モジュールで冪乗演算を行っていることを解析できた場合を考える。このケースは、すなわち、逆変換モジュールのみが解析者により解析困難である場合である。この場合においても、ｎの大きさが素因数分解が困難なぐらい、例えば、１０２４ビットぐらいであれば、ｎの素因数分解結果であるｐ，ｑを得ることが困難になる。また、ｐ，ｑが得られなければ、剰余整数環Ｚ／ｎＺの乗法群における離散対数問題を解くことが困難になる。一般に乗法群の大きさ（元の数）が１０２４ビットの数のように大きい場合は、それ上の離散対数問題も困難になる。実施の形態２では、ｐが既知の場合は逆変換モジュールにおける逆変換の方法によって、Ｚ／ｐ＾２　Ｚの乗法群における離散対数問題を容易に解けるようになる。実施の形態２における変換はこのように、ｐが既知であれば逆変換が容易であるが、既知でなければ困難であることを利用している点が、実施の形態１と異なる。
【００４４】
〔実施の形態３〕
本発明にかかる実施の形態３としての加算プログラム３００について説明する。本加算

プログラム３００は、楕円曲線のスカラ倍演算を利用している。楕円曲線については、非特許文献３が詳しい。

図３は、実施の形態３における加算プログラム３００の構成を示す図である。本プログラムは、実施の形態１と同様の出力モジュール１４０と、実施の形態１と異なる変換モジュール３１０、主要演算モジュール３２０、逆変換モジュール３３０から構成される。本プログラムは、入力ａ，ｂに対し、ａ＋ｂを出力するものである。

【００４５】

＜各種パラメータ・記号の定義、入力の条件＞

ここで、実施の形態３としての加算プログラム３００で使用する各種パラメータ・記号の定義を示す。

ｐ，ｑを素数とし、ｎ＝ｐ×ｑとする。ｐ，ｑは逆変換モジュール３３０が保持し、ｎは変換モジュール３１０、主要演算モジュール３２０がそれぞれ保持する。

【００４６】

楕円曲線Ｅの方程式をｙ＾２＝ｘ＾３＋Ａ×ｘ＋Ｂとする。Ａ，Ｂはパラメータである。Ｇ＝（ｘｇ，ｙｇ）　ｍｏｄ　ｎを楕円曲線Ｅ上の点とする。すなわち、ｙｇ＾２＝ｘｇ＾３＋Ａ×ｘｇ＋Ｂ　ｍｏｄ　ｎを満たす。Ａ，Ｂ，Ｇは、変換モジュール３１０、主要演算モジュール３２０、逆変換モジュール３３０が保持する。

楕円曲線Ｅの方程式をもつ体ＧＦ（ｐ）上の楕円曲線の点から構成される群をＥ（ＧＦ（ｐ））と書く。同様に、楕円曲線Ｅの方程式をもつ体ＧＦ（ｑ）上の楕円曲線の点から構成さえる群をＥ（ＧＦ（ｑ））と書く。Ｚ／ｎＺ上の楕円曲線の群をＥ（ＧＦ（ｐ））とＥ（ＧＦ（ｑ））の直積Ｅ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））で表す。Ｚ／ｎＺは体ではなく、環であるため、数学的には楕円曲線とはよべないが、ここでは便宜上、その直積Ｚ／ｎＺ上の楕円曲線の群とよぶ。

【００４７】

Ｅ（ＧＦ（ｐ））上の点Ｇｐ＝（ｘｇｐ，ｙｇｐ）　ｍｏｄ　ｐと、Ｅ（ＧＦ（ｑ））上の点Ｇｑ＝（ｘｇｑ，ｙｇｑ）　ｍｏｄ　ｑに対応するＺ／ｎＺ上の楕円曲線Ｅ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））の点Ｇ＝（ｘｇ，ｙｇ）　ｍｏｄ　ｎは、以下のように定義する。ｘｇをｘｇ　ｍｏｄ　ｐ＝ｘｇｐ，ｘｇ　ｍｏｄ　ｑ＝ｘｇｑを満たす数、ｙｇをｙｇ　ｍｏｄ　ｐ＝ｙｇｐ，ｘｇ　ｍｏｄ　ｑ＝ｘｇｑを満たす数とする。　この定義より、Ｅ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））上の点Ｇ＝（ｘｇ，ｙｇ）　ｍｏｄ　ｎに対応するＥ（ＧＦ（ｐ））上の点ＧｐをＧｐ＝（ｘｇｐ，ｙｇｐ）　ｍｏｄ　ｐとし、Ｅ（ＧＦ（ｑ））上の点ＧｑをＧｑ＝（ｘｇｑ，ｙｇｑ）とすることで、Ｅ（ＧＦ（ｐ）），Ｅ（ＧＦ（ｑ））をＥ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））の部分群とみなす。

【００４８】

実施の形態３においては、上記楕円曲線Ｅはｍｏｄ　ｐでの楕円曲線の位数、すなわち、点の個数がｐであるとする。このような体ＧＦ（ｐ）上の楕円曲線をアノマラス（Ａｎｏｍａｌｏｕｓ）楕円曲線とよぶ。さらに、ｍｏｄ　ｑでの楕円曲線の位数がｑである、すなわち、ＧＦ（ｑ）上でもアノマラス楕円曲線であるとする。このとき、Ｚ／ｎＺ上の楕円曲線は、スーパーアノマラス（Ｓｕｐｅｒ－Ａｎｏｍａｌｏｕｓ）楕円曲線とよぶ。スーパーアノマラス楕円曲線については非特許文献４が詳しい。このとき、Ｚ／ｎＺ上の楕円曲線の群はＥ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））であるので、楕円曲線の位数はｎ（＝ｐ×ｑ）となる。

【００４９】

入力ａ，ｂは、ｐ／２より小さい非負整数とする。

＜変換モジュール３１０の構成・処理＞

変換モジュール３１０は、ｎ，Ａ，Ｂ，Ｇを格納する第１のパラメータ格納手段と、入力ａ，ｂに対し、第１のパラメータ格納手段に格納されているｎ，Ａ，Ｂ，Ｇを用いて、Ｇａ＝ａ＊Ｇ　ｍｏｄ　ｎ，Ｇｂ＝ｂ＊Ｇ　ｍｏｄ　ｎを計算するスカラ倍演算手段を備える。ここで、ａ＊Ｇは、Ｇをａ回楕円曲線の加算により足し合わせた点である。また、ａ＊Ｇ　ｍｏｄ　ｎとは、ａ＊Ｇの各座標をｍｏｄ　ｎしたものである。

【００５０】
変換モジュール３１０は、入力ａ，ｂに対し、Ｇａ＝ａ＊Ｇ　ｍｏｄ　ｎ，Ｇｂ＝ｂ＊Ｇ　ｍｏｄ　ｎを計算し、それらの計算結果Ｇａ，Ｇｂを主要演算モジュール３２０へ入力する。
＜主要演算モジュール３２０の構成・処理＞
主要演算モジュール３２０は、ｎ，Ａ，Ｂを格納する第２のパラメータ格納手段と、入力Ｇａ，Ｇｂに対し、Ｇａｂ＝Ｇａ＋Ｇｂ　ｍｏｄ　ｎを計算する楕円曲線加算手段を備える。
【００５１】
主要演算モジュール３２０は、入力Ｇａ，Ｇｂに対し、第２のパラメータ格納手段に格納されているｎ，Ａ，Ｂを用いて、楕円曲線加算を実行し、Ｇａｂ＝Ｇａ＋Ｇｂ　ｍｏｄ　ｎを計算し、その計算結果Ｇａｂを逆変換モジュール３３０へ入力する。
＜逆変換モジュール３３０の構成・処理＞
逆変換モジュール３３０は、ｐ，Ａ，Ｂ，Ｇ　ｍｏｄ　ｐを格納する第３のパラメータ格納手段と、入力Ｇａｂに対し、第３のパラメータ格納手段に格納されているｐを用いて、Ｇｐ＝Ｇａｂ　ｍｏｄ　ｐ　を計算する還元手段と、Ｇ　ｍｏｄ　ｐに対するＧｐの離散対数を計算する楕円離散対数計算手段を備える。
【００５２】
逆変換モジュール３３０は、まず、入力Ｇａｂに対し、Ｇｐ＝Ｇａｂ　ｍｏｄ　ｐを計算する。次に、Ｇ　ｍｏｄ　ｐに対するＧｐの離散対数ｃ　ｍｏｄ　ｐを計算し、ｃを出力モジュール１４０へ入力する。
楕円離散対数計算手段におけるｃは、アノマラス楕円曲線上の離散対数問題の解である。アノマラス楕円曲線上の離散対数問題を解く方法は、非特許文献３の８８～９１ページが詳しい。計算方法はこの文献に記載されているため、ここでは説明を省略する。
【００５３】
＜実施の形態３の全体の動作＞
実施の形態３における加算プログラム３００の全体の動作を以下に示す。
加算プログラム３００の変換モジュール３１０は、入力ａ，ｂに対しＧａ，Ｇｂを計算する。次に主要演算モジュール３２０は、Ｇａ，Ｇｂに対し、Ｇａｂを計算する。さらに、逆変換モジュール３３０は、Ｇａｂに対し、ｃを計算し、出力モジュール１４０は、ｃを出力する。
【００５４】
＜実施の形態３の動作検証＞
以下で、実施の形態３における加算プログラム３００が、入力ａ，ｂに対し、ａ＋ｂを出力していることを検証する。
変換モジュール３１０において、ａ，ｂに対し、Ｇａ＝ａ＊Ｇ　ｍｏｄ　ｎ，Ｇｂ＝ｂ＊Ｇ　ｍｏｄ　ｎを計算し、主要演算モジュール３２０で、Ｇａｂ＝Ｇａ＋Ｇｂ　ｍｏｄ　ｎを計算する。このとき、Ｇａｂ＝（ａ＋ｂ）＊Ｇ　ｍｏｄ　ｎを満たすことは明らかである。逆変換モジュール３３０では、まず、Ｇｐ＝Ｇａｂ　ｍｏｄ　ｐを計算し、Ｇ　ｍｏｄ　ｐに対するＧｐの離散対数ｃを求める。すなわち、Ｇｐ＝ｃ＊Ｇ　ｍｏｄ　ｐが成り立つ。したがって、ｃ＝ａ＋ｂ　ｍｏｄ　ｐであり、ａ＜ｐ／２，ｂ＜ｐ／２より、ａ＋ｂ＜ｐであるため、加算プログラム３００は、ａとｂの加算結果ａ＋ｂを出力していることになる。
【００５５】
＜実施の形態３の効果＞
実施の形態３における加算プログラム３００は、実施の形態１、２と同様に、加算を行う値を変換しており、変換モジュール３１０及び逆変換モジュール３３０が解析困難な場合に、変換後の値から変換前の値を推測することは困難である。さらに、加算プログラム３００は、主要演算モジュール３２０において、楕円曲線加算を行っており、この楕円曲線加算という演算から加算プログラム３００が整数の加算を実現していることを推測する

ことは困難である。したがって、整数の加算を行う入力の値の隠蔽だけではなく、整数の加算という演算自体も隠蔽できることになり、本実施の形態３は有効である。

【００５６】

実施の形態３における加算プログラム３００では、Ｚ／ｎＺ上の楕円曲線の群Ｅ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））のスカラ倍演算を変換モジュールで行い、その部分群であるＥ（ＧＦ（ｐ））における離散対数問題を逆変換モジュールで解いている。ここで、もし、解析者がｐ，ｑは分からないが、変換モジュールで冪乗演算を行っていることを解析できた場合を考える。このケースは、すなわち、逆変換モジュールのみが解析者により解析困難である場合である。この場合においても、ｎの大きさが素因数分解が困難なぐらい、例えば、１０２４ビットぐらいであれば、ｎの素因数分解結果であるｐ，ｑを得ることが困難になる。また、ｐ，ｑが得られなければ、Ｚ／ｎＺ上の楕円曲線の群Ｅ（ＧＦ（ｐ））×Ｅ（ＧＦ（ｑ））における離散対数問題を解くことが困難になる。一般に群の大きさ（元の数）が１０２４ビットの数のように大きい場合は、それ上の離散対数問題も困難になる。実施の形態３では、ｐが既知の場合は逆変換モジュールにおける逆変換の方法によって、楕円曲線の群Ｅ（ＧＦ（ｐ））における離散対数問題を容易に解けるようになる。実施の形態３における変換はこのように、ｐが既知であれば逆変換が容易であるが、既知でなければ困難であることを利用している点が、実施の形態１と異なる。

【００５７】

〔実施の形態４〕

本発明に係る実施の形態４としての暗号化プログラム４００について説明する。本暗号化プログラムは、特許文献３に記載されている暗号化方法に対し、鍵と平文または、暗号化の途中結果との加算の部分で、実施の形態１の加算プログラムを適用したものである。

図４は、実施の形態４における暗号化プログラム４００の構成を示す図である。本プログラムは、拡張鍵生成モジュール４１０と、鍵加算モジュール４２０と、ローテーションモジュールＡ４３０と、ローテーションモジュールＢ４４０と、ローテーションモジュールＣ４５０と、ローテーションモジュールＤ４６０と、排他的論理和モジュール４７０と、ビット分割モジュール４８０と、出力モジュール４９０を備える。

【００５８】

暗号化プログラム４００は、６４ビットの鍵Ｋと、６４ビットの平文Ｍに対し、６４ビットの暗号文Ｃを出力する。

＜拡張鍵生成モジュール４１０の処理＞

拡張鍵生成モジュール４１０は、鍵Ｋを用いて、３２ビットの８個の拡張鍵Ｋ１，Ｋ２，Ｋ３，…，Ｋ８を生成し、出力する。生成する方法については、特許文献３に記載されているため、説明を省略する。

【００５９】

＜鍵加算モジュール４２０の処理＞

鍵加算モジュール４２０は、実施の形態１の加算プログラムにより入力された２つの数の加算を実行し、加算結果を出力する。

＜ローテーションモジュールＡ４３０の処理＞

ローテーションモジュールＡ４３０は、入力Ｘに対し、Ｒｏｔ２（Ｘ）＋Ｘ＋１を計算し、出力する。ここで、Ｒｏｔ２は、左へ２ビット循環シフトすることを示す。

【００６０】

＜ローテーションモジュールＢ４４０の処理＞

ローテーションモジュールＢ４４０は、入力Ｘに対し、Ｒｏｔ４（Ｘ）　ＸＯＲ　Ｘを計算し、出力する。ここで、Ｒｏｔ４は、左へ４ビット循環シフトすることを示し、ＸＯＲ排他的論理和を示す。

＜ローテーションモジュールＣ４５０の処理＞

ローテーションモジュールＣ４５０は、入力Ｘに対し、Ｒｏｔ８（Ｘ）　ＸＯＲ　Ｘを計算し、出力する。ここで、Ｒｏｔ８は左へ８ビット循環シフトすることを示す。

【００６１】

＜ローテーションモジュールＤ４６０の処理＞
ローテーションモジュールＤ４６０は、入力Ｘ，Ｙに対し、Ｒｏｔ１６（Ｘ）＋（ＸＡＮＤ　Ｙ）を計算し、出力する。ここで、Ｒｏｔ１６は、左へ１６ビット循環シフトすることを示し、ＡＮＤは論理積を示す。
＜排他的論理和モジュール４７０の処理＞
排他的論理和モジュール４７０は、入力Ｘ，Ｙに対し、Ｘ　ＸＯＲ　Ｙを計算し、出力する。
【００６２】
＜ビット分割モジュール４８０の処理＞
ビット分割モジュール４８０は、入力である６４ビットの数Ｘに対し、Ｘの上位ビット３２ビットＸ１、下位ビット３２ビットＸ２を出力する。
＜出力モジュール４９０の処理＞
出力モジュール４９０は、入力である３２ビットの数Ｘ，Ｙに対し、Ｘを上位、Ｙを下位とする６４ビット整数を生成し、プログラムの外部へ出力する。
【００６３】
＜暗号化プログラム４００の処理＞
暗号化プログラム４００の処理を以下に示す。図５は、暗号化プログラム４００の処理を示すフローチャートである。
ステップＳ１０１：拡張鍵生成モジュール４１０は、入力の鍵ＫからＫ１，Ｋ２，…，Ｋ８を生成する。
【００６４】
ステップＳ１０２：分割モジュール４９０は、入力の平文Ｍを上位Ｍ１と下位Ｍ２に分割する。
ステップＳ１０３：排他的論理和モジュール４７０は、Ｍ１とＭ２に対し、ＴＭＰ１＝Ｍ１　ＸＯＲ　Ｍ２を計算する。
ステップＳ１０４：鍵加算モジュール４２０は、ＴＭＰ１，Ｋ１に対し、ＴＭＰ２＝ＴＭＰ１＋Ｋ１を計算する。
【００６５】
ステップＳ１０５：ローテーションモジュールＡ４３０は、ＴＭＰ２に対し、ＴＭＰ３＝Ｒｏｔ２（ＴＭＰ２）＋ＴＭＰ２＋１を計算する。
ステップＳ１０６：ローテーションモジュールＣ４５０は、ＴＭＰ３に対し、ＴＭＰ４＝Ｒｏｔ４（ＴＭＰ３）　ＸＯＲ　ＴＭＰ３を計算する。
ステップＳ１０７：排他的論理和モジュール４７０は、ＴＭＰ４，Ｍ１に対し、ＴＭＰ５＝ＴＭＰ４　ＸＯＲ　Ｍ１を計算する。
【００６６】
ステップＳ１０８：鍵加算モジュール４２０は、ＴＭＰ５，Ｋ２に対し、ＴＭＰ６＝ＴＭＰ５＋Ｋ２を計算する。
ステップＳ１０９：ローテーションジュールＡ４３０は、ＴＭＰ６に対し、ＴＭＰ７＝Ｒｏｔ２（ＴＭＰ６）＋ＴＭＰ６＋１を計算する。
ステップＳ１１０：ローテーションモジュールＢ４４０は、ＴＭＰ７に対し、ＴＭＰ８＝Ｒｏｔ８（ＴＭＰ７）＋ＴＭＰ７＋１を計算する。
【００６７】
ステップＳ１１１：鍵加算モジュール４２０は、ＴＭＰ８とＫ３に対し、ＴＭＰ９＝ＴＭＰ８＋Ｋ３を計算する。
ステップＳ１１２：ローテーションモジュールＡ４３０は、ＴＭＰ９に対し、ＴＭＰ１０＝Ｒｏｔ２（ＴＭＰ９）＋ＴＭＰ９＋１を計算する。
ステップＳ１１３：ローテーションモジュールＤ４６０は、ＴＭＰ１０とＴＭＰ７に対し、ＴＭＰ１１＝Ｒｏｔ１６（ＴＭＰ１０）＋（ＴＭＰ１０　ＡＮＤ　ＴＭＰ７）を計算する。
【００６８】

ステップＳ１１４：排他的論理和モジュール４７０は、ＴＭＰ１１とＴＭＰ１に対して、ＴＭＰ１２＝ＴＭＰ１１　ＸＯＲ　ＴＭＰ１を計算する。
ステップＳ１１５：鍵加算モジュール４２０は、ＴＭＰ１２とＫ４に対し、ＴＭＰ１３＝ＴＭＰ１２＋Ｋ４を計算する。
ステップＳ１１６：ローテーションモジュールＡ４３０は、ＴＭＰ１３に対し、ＴＭＰ１４＝Ｒｏｔ２（ＴＭＰ１３）＋ＴＭＰ１３＋１を計算する。
【００６９】
ステップＳ１１７：排他的論理和モジュール４７０は、ＴＭＰ１４とＴＭＰ４に対し、ＴＭＰ１５＝ＴＭＰ１４　ＸＯＲ　ＴＭＰ４を計算する。
ステップＳ１１８：排他的論理和モジュール４７０は、ＴＭＰ１５とＴＭＰ１２に対し、ＴＭＰ１６＝ＴＭＰ１５　ＸＯＲ　ＴＭＰ１２を計算する。
ステップＳ１１９：出力モジュール４９０は、ＴＭＰ１５を上位、ＴＭＰ１６を下位とする６４ビットの整数を生成し、暗号文Ｃとして出力する。
【００７０】
＜実施の形態４の効果＞
実施の形態４における暗号化プログラムでは、鍵と平文または、暗号化の途中結果との加算において、実施の形態１の加算プログラムを利用する。そのため、加算を行う値、すわなち、鍵の値を推測することが困難になる。また、解析者が暗号アルゴリズムを知っている場合においても、鍵加算部分が、鍵との「加算」を行っていると推測しにくい。そのため、解析者が、暗号アルゴリズムの特徴である鍵加算部分をプログラム内から探し出す攻撃をした場合でも、鍵加算部分を探し出すことが困難なため、攻撃成功も困難となる。このように、解析者の攻撃が困難になり、本実施の形態４は有効である。
【００７１】
（変形例）
上記に説明した実施の形態は、本発明の実施の一例であり、本発明はこの実施の形態に何ら限定されるものではなく、その旨を逸脱しない範囲において種々なる態様で実施し得るものである。例えば、以下のような場合も本発明に含まれる。
（１）実施の形態１、２、３において、２個の非負整数ａ，ｂの加算を行っていたが、これを３個またはそれ以上の個数の非負整数の加算を行ってもよい。その場合は、変換モジュールはそれぞれの非負整数を変換し、主要演算モジュールでは、それぞれの変換結果を実施の形態１、２では乗算、実施の形態３では楕円曲線加算を行う。
【００７２】
（２）暗号化プログラムにおいて、鍵加算部分に実施の形態１の加算プログラムを使用したが、これが実施の形態２または３の加算プログラムであってもよい。
（３）暗号化プログラムにおいて、鍵加算部分のみに加算プログラムを使用したが、他の加算部分についても使用してもよい。
（４）実施の形態４では、加算プログラムを暗号化プログラムに適用しているが、復号化プログラムや、デジタル署名を生成する署名生成プログラムに適用するとしてもよい。
【００７３】
（５）実施の形態１、２、３においては、剰余整数環の乗法群、楕円曲線上の群を利用したが、これを、その他の群を利用するとしてもよい。また、実施の形態１、２では冪乗演算、実施の形態３においては、楕円曲線のスカラ倍演算を行って、整数を変換したが、これをその他の群の冪演算としてもよい。冪演算とは、群の基本演算、すなわち、剰余整数環では乗算、楕円曲線上の群では楕円曲線加算を、整数回行った結果を求める演算である。したがって、剰余整数環の乗法群の冪演算は冪乗演算、楕円曲線上の群の冪演算は楕円曲線のスカラ倍演算である。実施の形態２では、剰余整数環Ｚ／ｎＺの乗法群の「部分群」である、剰余整数環Ｚ／ｐ＾２　Ｚの乗法群において離散対数問題を解いている。その他の群を使用する場合は、実施の形態２と同様に逆変換モジュールでその他の群の「部分群」において離散対数問題を解いてもよい。
【００７４】

（６）実施の形態１において、ｇはｍｏｄ　ｐｉ（ｉ＝１，２，…，ｋ）において原始元としたが、原始元でなくてもよい。その場合は、ｇ＾ｍｉ＝１　ｍｏｄ　ｐｉ（ｍｉ＞０）となるｍｉに対し、Ｌ＝ｍ１×ｍ２×…×ｍｋとする。

（７）これらの実施の形態及び変形例の組合せであってもよい。

【産業上の利用可能性】

【００７５】

これらの構成によると、演算に使用する値の隠蔽だけでなく、演算そのものを隠蔽することができる。したがって、本技術を用いた難読化ソフトウェアをＩＣカード等の機器に組み込むことは有用である。

【図面の簡単な説明】

【００７６】

【図１】本発明に係る１個の実施の形態としての加算プログラム１００の構成を示すブロック図

【図２】本発明に係る１個の実施の形態としての加算プログラム２００の構成を示すブロック図

【図３】本発明に係る１個の実施の形態としての加算プログラム３００の構成を示すブロック図

【図４】本発明に係る１個の実施の形態としての暗号化プログラム４００の構成を示すブロック図

【図５】暗号化プログラム４００の処理を示すフローチャート

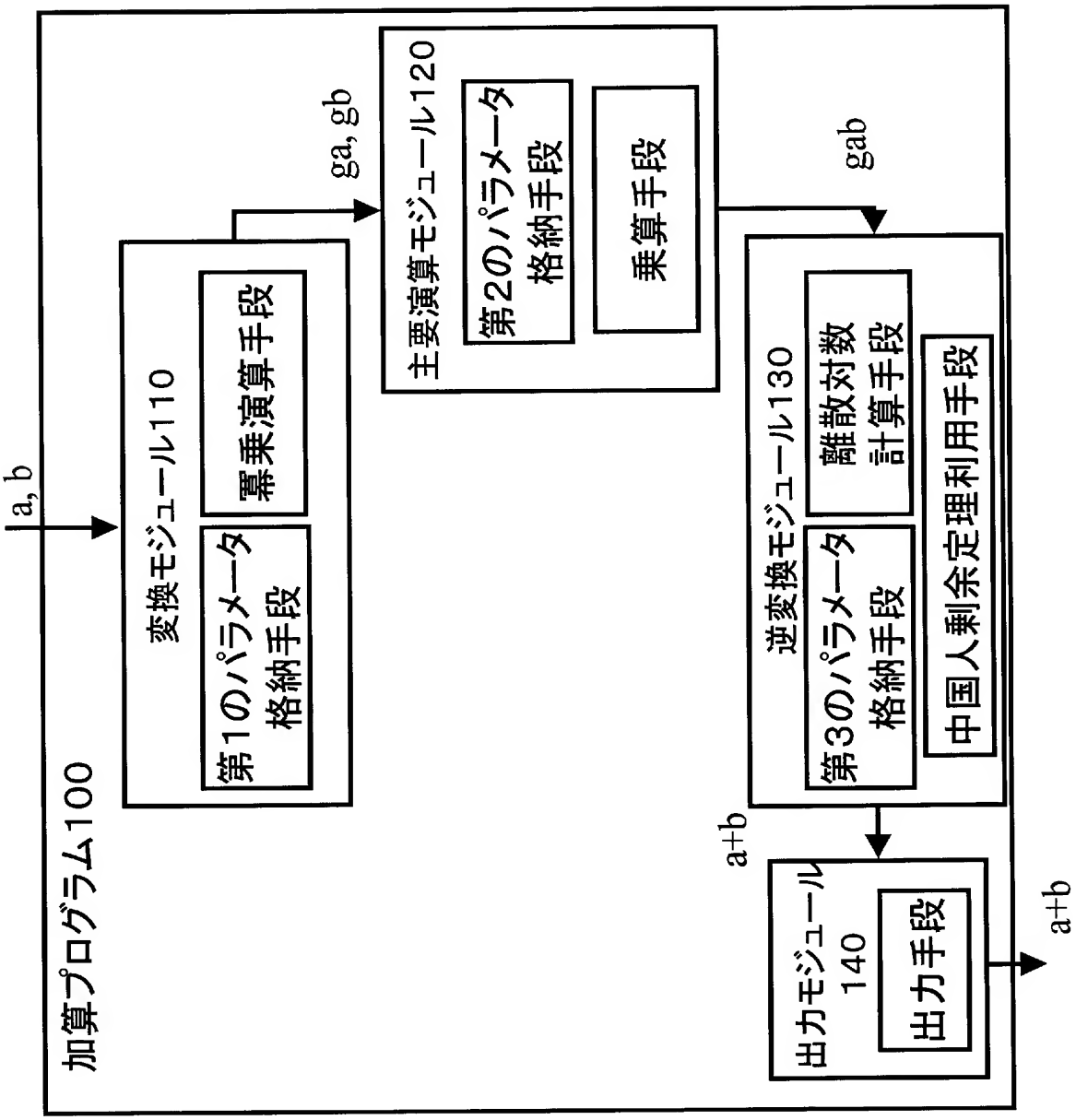
【図６】従来例としての加算プログラム５００の構成を示すブロック図

【符号の説明】

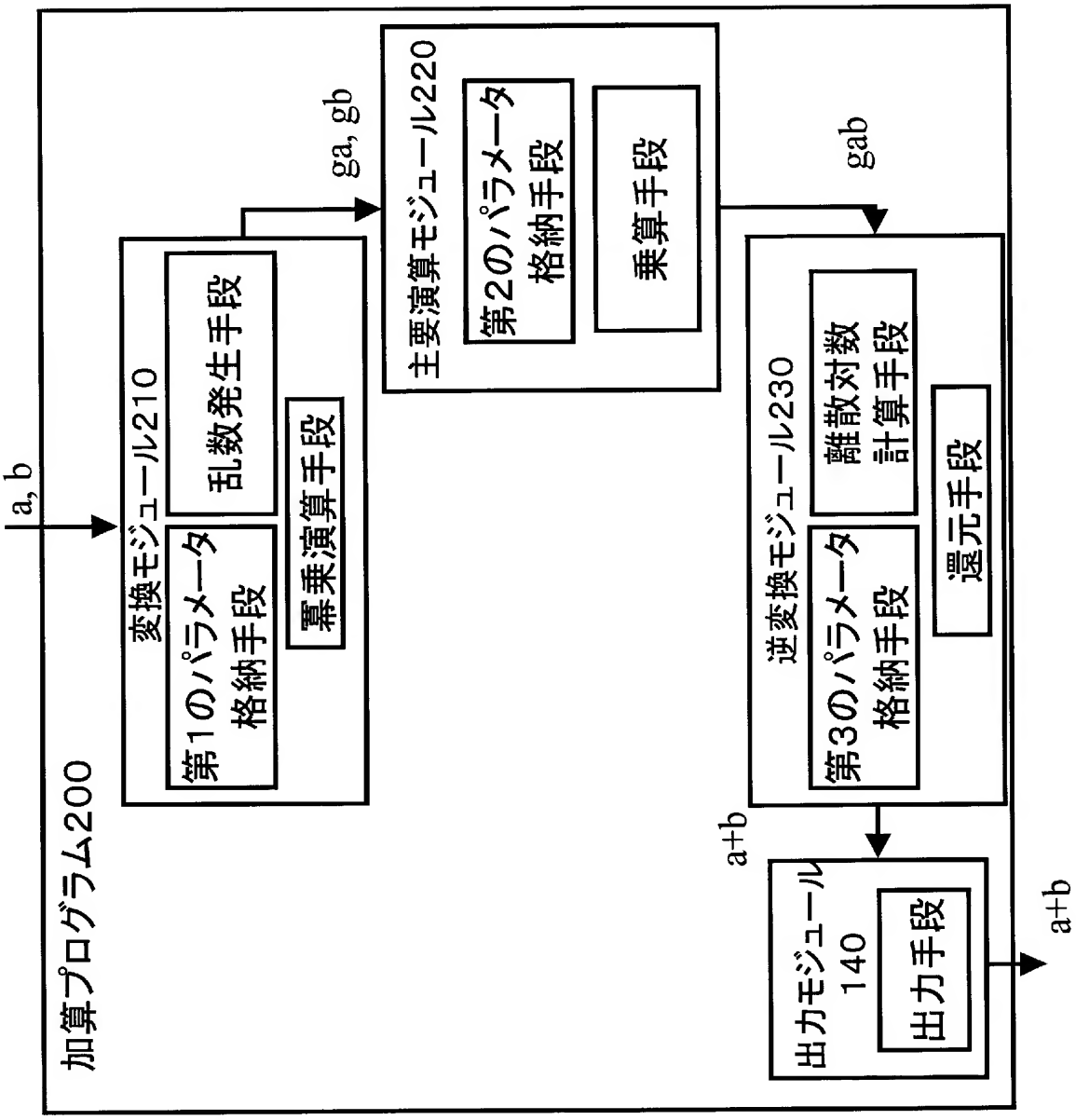
【００７７】

１００、２００、３００、５００　加算プログラム
１１０、２１０、３１０、５１０　変換モジュール
１２０、２２０、３２０、５２０　主要演算モジュール
１３０、２３０、３３０、５３０　逆変換モジュール
１４０、４９０、５４０　出力モジュール
４００　暗号化プログラム
４１０　拡張鍵生成モジュール
４２０　鍵加算モジュール
４３０　ローテーションモジュールＡ
４４０　ローテーションモジュールＢ
４５０　ローテーションモジュールＣ
４６０　ローテーションモジュールＤ
４７０　排他的論理和モジュール
４８０　ビット分割モジュール

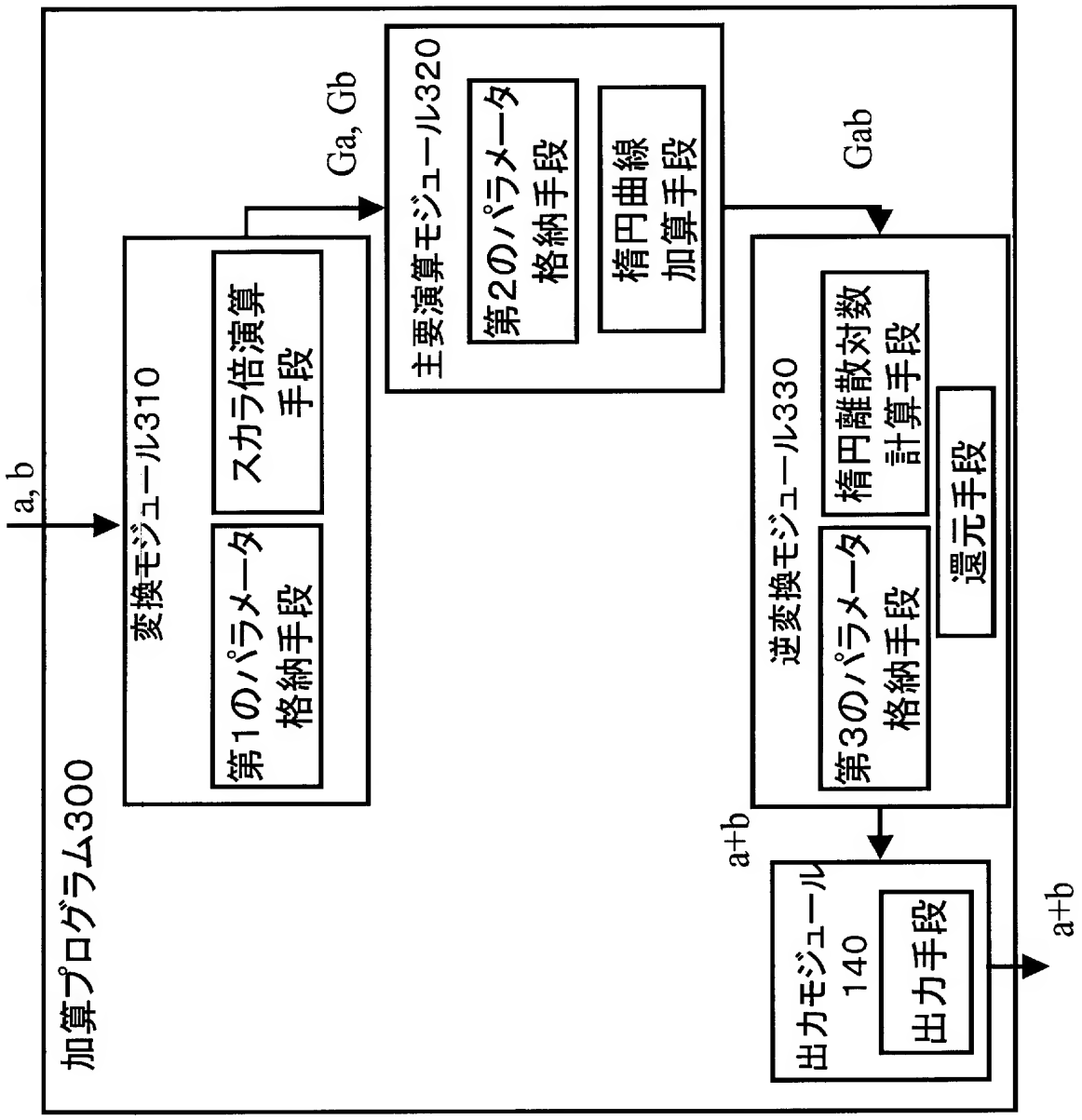
【書類名】図面
【図１】


加算プログラム100
a, b
変換モジュール110
第1のパラメータ格納手段
冪乗演算手段
ga, gb
主要演算モジュール120
第2のパラメータ格納手段
乗算手段
gab
逆変換モジュール130
第3のパラメータ格納手段
離散対数計算手段
中国人剰余定理利用手段
a+b
出力モジュール140
出力手段
a+b

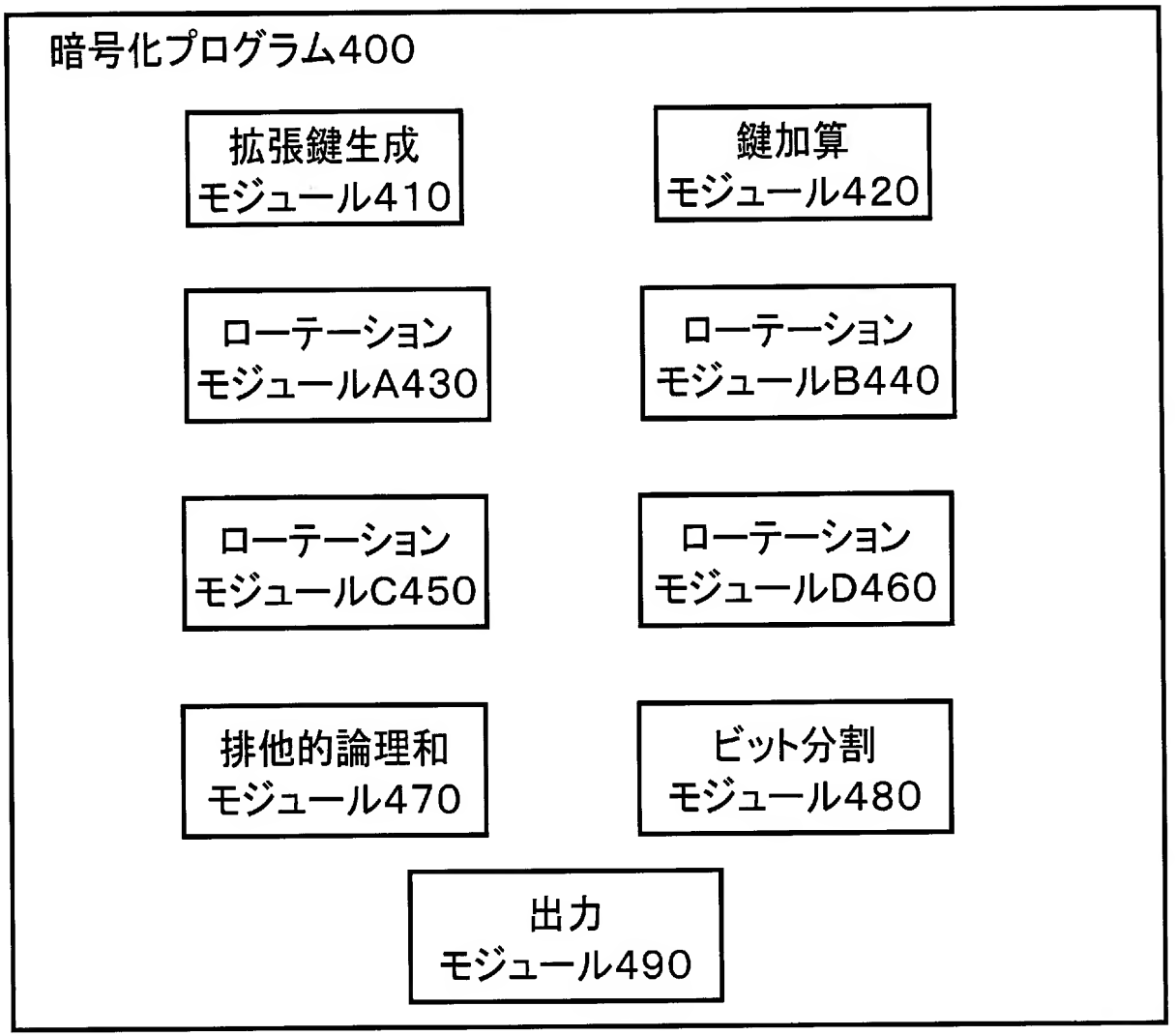
【図２】


加算プログラム200
a, b
変換モジュール210
第1のパラメータ格納手段
乱数発生手段
冪乗演算手段
ga, gb
主要演算モジュール220
第2のパラメータ格納手段
乗算手段
gab
逆変換モジュール230
第3のパラメータ格納手段
離散対数計算手段
還元手段
a+b
出力モジュール140
出力手段
a+b

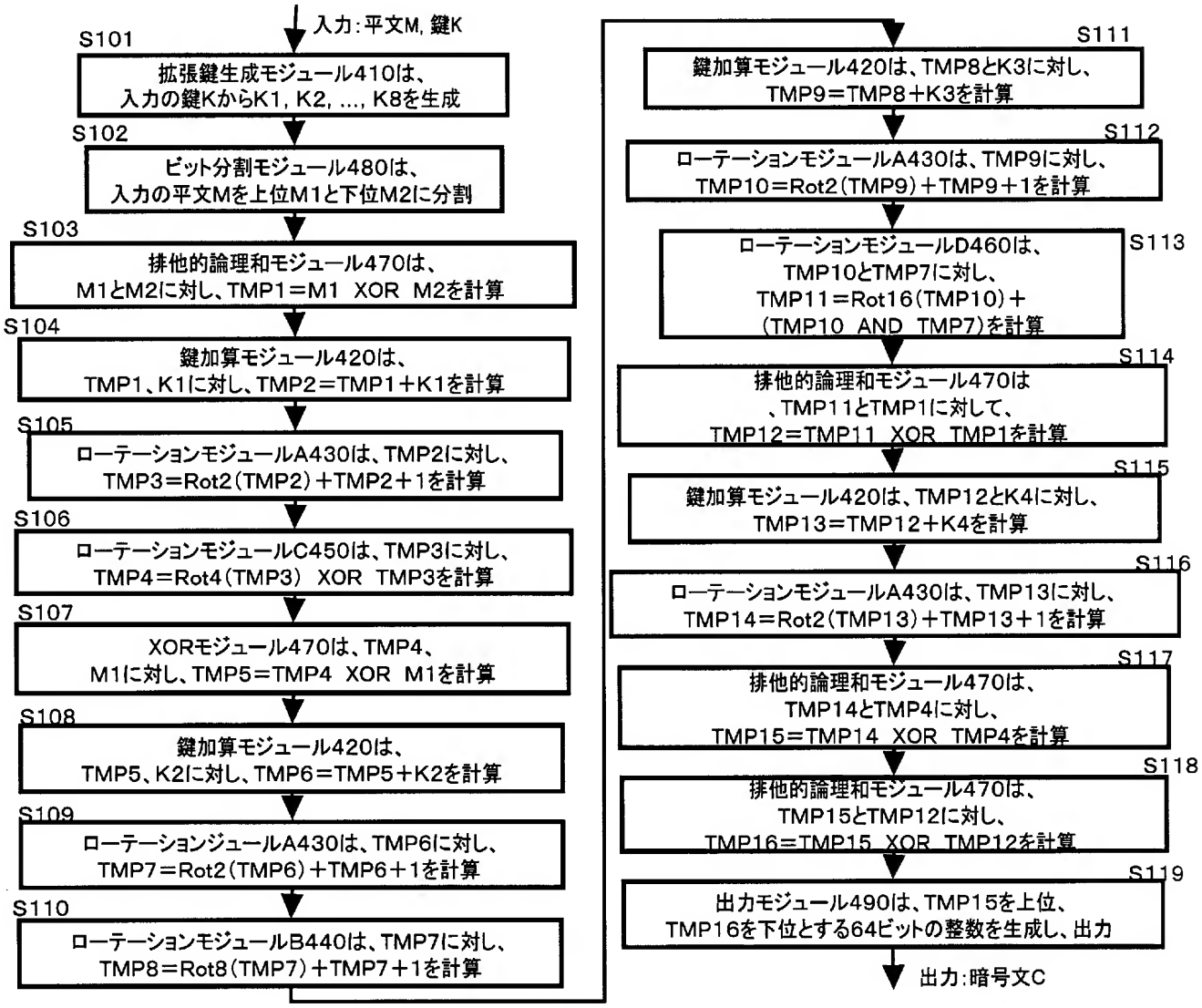
【図３】


加算プログラム300
a, b
変換モジュール310
第1のパラメータ格納手段
スカラ倍演算手段
Ga, Gb
主要演算モジュール320
第2のパラメータ格納手段
楕円曲線加算手段
Gab
逆変換モジュール330
第3のパラメータ格納手段
楕円離散対数計算手段
還元手段
a+b
出力モジュール140
出力手段
a+b

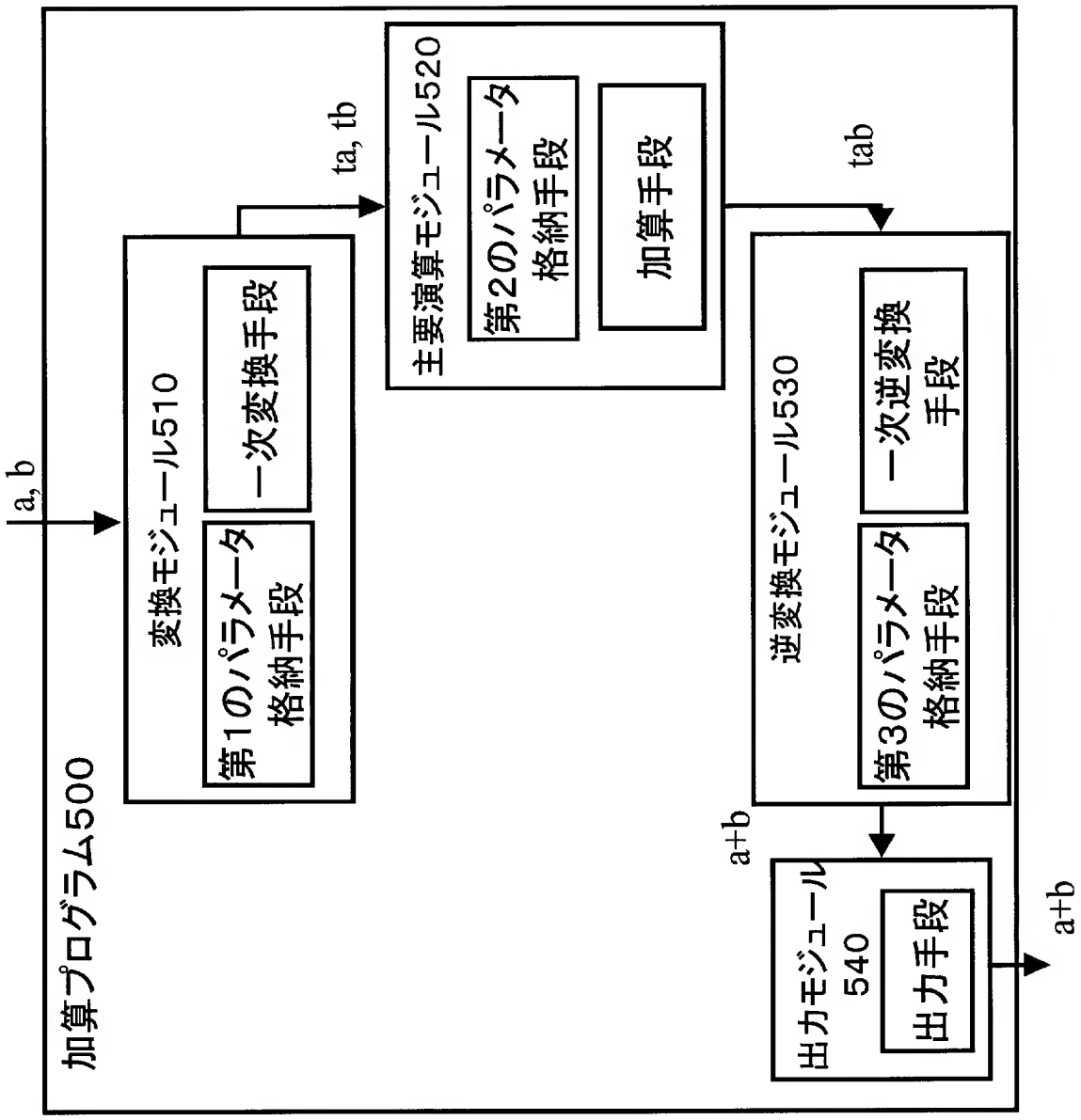
【図４】


暗号化プログラム400
拡張鍵生成
モジュール410
鍵加算
モジュール420
ローテーション
モジュールA430
ローテーション
モジュールB440
ローテーション
モジュールC450
ローテーション
モジュールD460
排他的論理和
モジュール470
ビット分割
モジュール480
出力
モジュール490

【図５】


入力:平文M, 鍵K
S101
拡張鍵生成モジュール410は、
入力の鍵KからK1, K2, ..., K8を生成
S102
ビット分割モジュール480は、
入力の平文Mを上位M1と下位M2に分割
S103
排他的論理和モジュール470は、
M1とM2に対し、TMP1＝M1 XOR M2を計算
S104
鍵加算モジュール420は、
TMP1、K1に対し、TMP2＝TMP1＋K1を計算
S105
ローテーションモジュールA430は、TMP2に対し、
TMP3＝Rot2(TMP2)＋TMP2＋1を計算
S106
ローテーションモジュールC450は、TMP3に対し、
TMP4＝Rot4(TMP3) XOR TMP3を計算
S107
XORモジュール470は、TMP4、
M1に対し、TMP5＝TMP4 XOR M1を計算
S108
鍵加算モジュール420は、
TMP5、K2に対し、TMP6＝TMP5＋K2を計算
S109
ローテーションジュールA430は、TMP6に対し、
TMP7＝Rot2(TMP6)＋TMP6＋1を計算
S110
ローテーションモジュールB440は、TMP7に対し、
TMP8＝Rot8(TMP7)＋TMP7＋1を計算
S111
鍵加算モジュール420は、TMP8とK3に対し、
TMP9＝TMP8＋K3を計算
S112
ローテーションモジュールA430は、TMP9に対し、
TMP10＝Rot2(TMP9)＋TMP9＋1を計算
S113
ローテーションモジュールD460は、
TMP10とTMP7に対し、
TMP11＝Rot16(TMP10)＋
(TMP10 AND TMP7)を計算
S114
排他的論理和モジュール470は
、TMP11とTMP1に対して、
TMP12＝TMP11 XOR TMP1を計算
S115
鍵加算モジュール420は、TMP12とK4に対し、
TMP13＝TMP12＋K4を計算
S116
ローテーションモジュールA430は、TMP13に対し、
TMP14＝Rot2(TMP13)＋TMP13＋1を計算
S117
排他的論理和モジュール470は、
TMP14とTMP4に対し、
TMP15＝TMP14 XOR TMP4を計算
S118
排他的論理和モジュール470は、
TMP15とTMP12に対し、
TMP16＝TMP15 XOR TMP12を計算
S119
出力モジュール490は、TMP15を上位、
TMP16を下位とする64ビットの整数を生成し、出力
出力:暗号文C

【図６】


加算プログラム500
a, b
変換モジュール510
第1のパラメータ格納手段
一次変換手段
ta, tb
主要演算モジュール520
第2のパラメータ格納手段
加算手段
tab
逆変換モジュール530
第3のパラメータ格納手段
一次逆変換手段
a+b
出力モジュール540
出力手段
a+b

【書類名】要約書
【要約】
【課題】従来の耐タンパーソフト技術として知られている演算の領域を変換する方式では、領域を変換することで値を隠蔽することができるが、何の演算を行っているかは解析者に分かるため、演算の種類に特徴がある場合はその部分を集中して解析される可能性があり望ましくない。そこで、演算そのものも変換する方式が望まれている。
【解決手段】　演算を整数加算とした場合に、被演算整数を冪乗演算を用いて剰余整数環の元に変換し、剰余整数環の乗算を行い、その結果を逆変換することにより加算を実現する。
【選択図】　図２

出願人履歴

０００００５８２１
19900828
新規登録

大阪府門真市大字門真１００６番地
松下電器産業株式会社